Wet Paper Coding for Watermarking of Binary Images

Michail Zubarev and Valery Korzhik
State University of Telecommunications
St. Petersburg
Russia
Email: korzhik@spb.lanck.net

Guillermo Morales-Luna
Computer Science
CINVESTAV-IPN
Mexico City
Mexico
Email: gmorales@cs.cinvestav.mx

Abstract—We propose a new method to embed data in binary images, including scanned text, figures, and signatures. Our method relies on the concept of wet paper codes. The shuffling before embedding is used in order to equalize irregular embedding capacity from diverse areas in the image. The hidden data can be extracted without the original binary image. We illustrate some examples of watermarked binary images after wet paper coding.

I. INTRODUCTION

A great majority of works on image watermarking (WM) has been devoted to color or gray scale images where pixel luminances may take values on a wide range. Then slight changings of even most of pixel luminance values may not be noticeable under normal viewing conditions.

Data hiding on digital binary images (BI) has been investigated in [1]. The main characteristic of this approach is the notion of *flippable pixels* which are precisely those that can be inverted (flipped) without noticeable corruption of BI. In our scheme, a bit string is embedded in each block based on the flippability score of the pixel within the block. This score is measured by the changes in smoothness and connectivity. A shuffling table or a key is required for correct extraction of the bit string watermark. In [2], Wu *et al.* have presented a flippability score computation method by considering the smoothness and connectivity of the cover image. In their work, the score is determined by considering the changes in smoothness and connectivity and it is reached by the proposed rules. However, the involved analysis is quite extensive for neighborhoods larger than 3×3 pixels. Another flippability score computation method was proposed in [3]. That method is based on the reciprocal distance calculation, which matches well within subjective evaluation, and it is easy to extend to block neighborhoods larger than 3×3 pixels. In [4], the authors use a weight matrix to calculate the new transitions created by the flipping. That approach calculates the distortion by combining the changes in smoothness and in connectivity. Maxemchuk *et al.* [5] changed line spacing and character spacing in order to embed information in textual images for bulk electronic publications. This method cannot be easily extended to other binary images and the size of hidden data is very small.

In all those approaches the embedding capacity is not large.
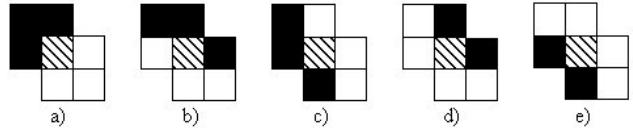
Fig. 1. 3×3 patterns with flippable center pixels.

In order to deal with this limitation, an approach has been proposed, called *writing on wet paper* [6].

In our proposed procedure, we exploit the notion of flippable pixels, as presented in [1], and we combine it with the *wet paper codes* (WPC) given in [6]. In section 2 we show a method for finding flippable pixels. Section 3 contains the description of the WPC technique. Section 4 contains some experimental results. Section 5 contains some conclusions and poses resulting open problems.

II. WM EMBEDDING METHOD FOR BI

It was mentioned already that it is rather difficult to embed a WM into a BI without noticeable distortion. In [1] an approach has been developed to look through each of the adjacent $N \times N$ areas of pixels and to embed a "0" in an area by either changing or not one of its pixels, so that the total number of black pixels in the area is an even number; similarly, to embed a "1", by either changing or not one pixel, so that the number of black pixels is an odd number. If it is necessary to change one pixel in an area it has to be chosen as a *flippable pixel*. Obviously the area should be non-empty. We will specify flippable pixels in greater detail.

Let us consider initially 3×3 areas with the intention to change only the central pixel (whenever it was necessary). In Figure 1 we present the patterns in which we will consider the central pixel as flippable, of course the central pixel of any pattern obtained by a rotation around the central axis or an inversion of those patterns is considered also as flippable. We remark that our choice of *central-flippable patterns* (CFP) is of a heuristic kind because we are mainly concerned in maintaining the connectivity features after a flipping. We may try to consider a larger area size, $N > 3$, but this will involve the same 3×3 patterns displayed at Figure 1. Namely, we will shift a 3×3 area, by one pixel in turn, in both horizontal and

vertical directions, along the $N \times N$ area until all flippable pixels are found.

III. WRITING OF WET PAPER FOR THE CASE OF BINARY IMAGES

This name derives from a real situation, when the paper has been exposed to rain. The sender is able to write the message only over dry places. During transmission the paper dries out and the recipient does not know, which places were dry when encoding. However he is able to read the message easily. In information theory this technique is known as *"writing in memory with defective cells"*.

A. Encoder

Let us assume that the sender has a binary column vector $\mathbf{b} = (b_i)_{i=1}^n$ received from a binary image of n pixels (a black pixel gives value 1, and a white pixel gives 0), and an index set $C \subset \{1, \dots, n\}$, $|C| = k$, corresponding to those bits that are flippable and subject to be modified in order to embed a message. The sender wants to communicate a sequence of q bits, $\mathbf{m} = (m_1, \dots, m_q)$. First, let us assume that the recipient knows q and let us postpone later the case when the recipient does not know q. The sender and recipient use a shared stegokey to generate a pseudo-random binary $(q \times n)$-matrix D. The sender shall modify some flippable bits b_i, $i \in C$, in such a way that the modified binary column vector $\mathbf{b}' = (b_i')_{i=1}^n$ satisfies $D\mathbf{b}' = \mathbf{m}$. Let $\mathbf{v} = \mathbf{b}' - \mathbf{b}$, hence the support of vector $\mathbf{v}$ is a subset of C, then

$$D\mathbf{v} = \mathbf{m} - D\mathbf{b} \tag{1}$$

Given $\mathbf{m}$ and $\mathbf{b}$, the system of linear equations (1) should be solved for k unknowns v_i, $i \in C$ (clearly, $v_i = 0$ whenever $i \notin C$). By deleting the rows and columns with indexes in the complement of C, the system (1) can be rewritten as

$$H\mathbf{v}' = \mathbf{m} - D\mathbf{b} \tag{2}$$

with H of order $(q \times k)$ and $\mathbf{v}' = (v_i)_{i \in C}$ a k-dimensional vector. The equations system (2) is posed in the Galois field GF(2) and it has an unique solution whenever $\text{rank}(H) = q$. Thus, in order to decide which flippable pixels should actually be flipped, the encoder should solve the equations system (2).

B. Decoder

The modified binary image $\mathbf{b}'$ is sent to the recipient. The decoding is very simple because the recipient may recover the message $\mathbf{m} = D\mathbf{b}'$ using the shared matrix D. Note that the recipient is not required to know which pixels are flippable.

We now explain how to relax the assumption that the recipient knows q. The sender and the recipient can agree matrix D in a row-by-row manner rather than as a whole planar $(q \times n)$-array of bits. The sender may reserve the first $\lceil \log_2 n \rceil$ bits of the message $\mathbf{m}$ as a header to inform the recipient of the number of rows in D (the symbol $\lceil x \rceil$ denotes the smallest integer greater or equal than x). The recipient takes the first $\lceil \log_2 n \rceil$ rows of D, and applies the resulting $(\lceil \log_2 n \rceil \times n)$-submatrix to the received vector $\mathbf{b}'$, in order to

	Image 1	Image 2	Image 3
N_A	16	16	16
N_{FP}	1283	5455	2358
N_E	1058	5222	2122

TABLE I
THE NUMBERS OF IMAGE AREAS N_A, FLIPPABLE PIXELS IN THE IMAGE N_{FP}, AND EMBEDDED BITS INTO AN IMAGE N_E.

recover the header, i.e. the message length q. Then, he uses the rest of D to recover the message $\mathbf{m} = D\mathbf{b}'$.

C. Practical implementation

Due to the great computational complexity of solving a system (2) for a big value of q, we partition our image into areas of N pixels, and for each we solve a corresponding system (2) of linear equations. Besides, since there is no a regular distribution of flippable pixels from area to area, we use shuffling of the image pixels as it was suggested in [7].

If the length of the embedded message is greater than the number of flippable pixels in the area, we try to embed as many bits as possible. The remaining bits would be embedded into following areas. As remarked before, whenever condition $\text{rank}(H) = q$ is fulfilled for an area, we may embed $k - \lceil \log_2 n \rceil$ bits in that area. Otherwise, we may decrease q until condition $\text{rank}(H) = q$ is satisfied.

IV. EXPERIMENTAL RESULTS

We took $n = 4096$ pixels as the area size. In Table I we present the numbers N_A, N_{FP}, N_E corresponding to image areas, flippable pixels in an image and embedded bits for three typical images.

The three processed images, with actual size of about 300×225 pixels, are presented in their original forms (before watermarking) and after watermarking in Figure 2.

V. CONCLUSIONS

It seems to be difficult the embedding into a BI of an additional information without remarkable distortions of the original image. But as it can be appreciated in Table I and Figure 2, it is indeed possible to embed a significant number of information bits without any visual difference among the original images and their watermarked copies.

It is an interesting open problem to find an efficient statistical method in order to distinguish the original BI from the watermarked BI. Solutions to this steganography problem are well known [8], [9] for gray-scale (or color) images but they have a poor adequacy for BI. Another interesting open problem is to develop a technique for *selective authentication* of BI when there is a group of legitimate and illegitimate distortions of the originals. If it occurs that distortions belong to the first group, then authentication is confirmed, while if it occurs at the second group, then authentication fails. The selective authentication is especially important for BI because the legitimate distortion can be resulted from channel noises or corruption after scanning of paper documents or images

(a)   (b)

Image 1

Despite the fact that computer ... decreased dramatically over t... data storage still remains, and wi... remain, an important cost factor f... database applications. Compressi... base system is attractive for two re... reduction and performance improv... duction is a direct and obvious be... mance improves because smaller a...

(a)   (b)
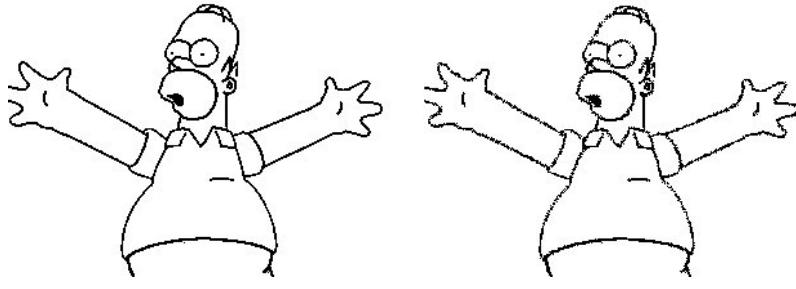
Image 2

(a)   (b)

Image 3

Fig. 2. Images 1, 2, 3 in (a) original and (b) after watermarking with WPC.

received by fax. We are going to present our contribution in this direction in the near future.

REFERENCES

[1] M. Wu, E. Tang, and B. Liu, "Data hiding in digital binary image," in *IEEE Int. Conf. on Multimedia & Expo (ICME'00)*. IEEE, 2000, pp. 31–45.
[2] M. Wu and B. Liu, "Data hiding in binary image for authentication and annotation." *IEEE Transactions on Multimedia*, vol. 6, no. 4, pp. 528–538, 2004.
[3] H. Lu, A. Kot, and Y. Shi, "Distance-reciprocal distortion measure for binary document images," *IEEE Signal Processing Letters*, vol. 11, no. 2, pp. 228–231, 2004.
[4] J. Cheng and A. Kot, "Objective distortion measure for binary images," in *TENCON 2004. 2004 IEEE Region 10 Conference*, 2004, pp. 355–358.
[5] N. F. Maxemchuk and S. H. Low, "Marking text documents." in *ICIP (3)*, 1997, pp. 13–13.
[6] J. J. Fridrich, M. Goljan, P. Lisonek, and D. Soukal, "Writing on wet paper," in *Security, Steganography, and Watermarking of Multimedia Contents*, ser. Proceedings of SPIE, E. J. Delp and P. W. Wong, Eds., vol. 5681. SPIE, 2005, pp. 328–340.
[7] M. Wu and B. Liu, "Digital watermarking using shuffling," in *ICIP (1)*, 1999, pp. 291–295.
[8] S. Damitrescu, X. Wu, and Z. Wang, "Detection of LSB steganography via sample pair analysis," in *International Workshop on Information Hiding 2002*, vol. 2578. Springer LNCS, 2002, pp. 355–372.
[9] J. Fridrich, M. Goljan, and D. Hogea, "Stegoanalysis of JPEG images: Breaking the F5 algorithm," in *Proc. IHW'2002*. Springer Lecture Notes in Computer Science 2578, 2002, pp. 310–323.